SWVL Mobile Challenge

A decade of Movies

The Challenge

The past decade held a lot of movies, some left a mark and some were just a set of 24-60 pictures per second. We would like you to create a **Master - Detail Application** to showcase those movies and the signature they left behind.

You will have a local list of movies that should be displayed in any order. The list is *searchable* and the search results will be *categorized* by **Year**.

Each search result category will hold at most the **top rated 5 movies** of this category (year**)**.

Once a movie is selected from the search results, you will switch to a detailed view to unveil the following:

- Movie Title
- Movie Year
- Movie Genres (if any)
- Movie Cast (if any)
- A two column list of pictures fetched from flickr that matches the movie title as the search query

```
{
    "title": "(500) Days of Summer",
    "year": 2009,
    "cast": [
        "Joseph Gordon-Levitt",
        "Zooey Deschanel"
    ],
    "genres": [
        "Romance",
        "Comedy"
    ],
    "rating": 4
}
```

Requirements

- Loading time of the local data isn't an issue, **searching** and **sorting** times should be optimized.
- You are free to play with any UI elements and kits, unleash the artist within
- You are free to use any libraries or third parties
- Use Github to host your project, and try to avoid committing all your work in one commit "Final Final Project Commit"
- Unit Tests is a must and UI Test is a bonus
- You are free to adopt any architecture you want
- Caching the loaded data to minimize the parsing time is a **bonus**

Before Starting

Before you kickstart your project, it's preferable that you go through the following:

- You've read and fully understood the requirements and if not don't hesitate to contact us for any extra information. We are here to ease the process as much as possible.
- You have your dataset attached with the task.

We expect these requirements to take around 6 - 8 hours, so no need to implement features that would take more time. However, this is not a speed test, take all the time you need.

Share with us the amount of time it took to complete the task, yet it will not be used for your evaluation.

Deliverables

- Github project url along with a released version
- A descriptive README file along with any extra configurations needed to run this app

Hints ;)

- Not all search terms are expected to return data from Flickr
- Try to have a solid and clean architecture for your project
- Unit tests.
- Comments. Share with us what you were thinking of

Flickr APIs

Flickr APIs are simple and straight forward. You have to create a key for the API authentication : https://www.flickr.com/services/api/misc.api_keys.html

You should use the Photos.Search API , The request should be something like this:

```
https://api.flickr.com/services/rest/?method=flickr.photos.search&api_key={YOUR_API_KEY}&format=json&nojsoncallback=1&text={MOVIE_TITLE}&page=1&per_page=10
```

This call should return a JSON list of the Flickr Picture Model that you will use to create the Picture url using the following url schema :

```
{
    "id": "32885403967",
    "owner": "27003603@N00",
    "secret": "be14b95a9c",
    "server": "65535",
    "farm": 66,
    "title": "Gabriel Ynoa",
    "ispublic": 1,
    "isfriend": 0,
    "isfamily": 0
}
```

```
http://farm{farm}.static.flickr.com/{server}/{id}_{secret}.jpg
```